

Zykov Kirill. Laboratory work №1

scenario: probability of mutation is 10 percent

Code:

```
namespace SIIT_1
{
    class Chromosome
    {
        private bool[] species { get; set; }
        private int length;
        private double fitnessValue;
        private static Random rnd = new Random(DateTime.Now.Millisecond);
        public Chromosome()
        {
            species = new bool[0];
            length = 0;
        }
        public Chromosome(int n)
        {
            species = new bool[n];
            for (int i = 0; i < n; i++)
            {
                int random = Rnd(rnd, 100);
                if (random > 50)
                {
                    species[i] = true;
                }
                else species[i] = false;
            }
            length = n;
        }
        public Chromosome OnePointCross(Chromosome partner, int point)
        {
            Chromosome chlid = new Chromosome(length);
            for (int i = 0; i < point; i++)
            {
                chlid.species[i] = species[i];
            }
            for (int i = point; i < length; i++)
            {
                chlid.species[i] = partner.species[i];
            }
            return chlid;
        }
        public void Mutation()
        {
            if (species[Rnd(rnd, species.Length)] == true) species[Rnd(rnd,
species.Length)] = false;
            else species[Rnd(rnd, species.Length)] = true;

        }
        public string GetSpecie()
        {
            string str = "";
            for (int i = 0; i < species.Length; i++)
            {
                str+=Convert.ToInt32(species[i]).ToString();
            }
            return str;
        }
        public int Rnd(Random r, int n)
    }
}
```

```

    {
        return r.Next(n);
    }
    public void EvaluateFitness()
    {
        fitnessValue = Convert.ToDouble(species.Count(ai => ai ==
true))/Convert.ToDouble(length);
    }
    public double GetFitnessValue()
    {
        EvaluateFitness();
        return fitnessValue;
    }
}
}

namespace SIIT_1
{
    class Population
    {
        private List<Chromosome> population;
        private int popSize;
        private int geneNum;
        public double avgFit;
        private static Random rnd = new Random(DateTime.Now.Millisecond);
        public Population()
        {
            population = new List<Chromosome>();
        }
        public Population(List<Chromosome> pop)
        {
            population = pop;
        }
        public void CreatePopulation(int size,int length)
        {
            popSize = size;
            for (int i = 0; i < size; i++)
            {
                Chromosome spec = new Chromosome(length);
                population.Add(spec);
            }
            geneNum = length;
        }
        public Population Selection()
        {
            Population newPop;
            population = SortPop().population;
            List<Chromosome> children= new List<Chromosome>();
            for (int i = 0; i < population.Count; i++)
            {
                children.Add(population[Rnd(rnd, population.Count /
2)].OnePointCross(population[Rnd(rnd, population.Count / 2)], Rnd(rnd, geneNum)));
            }
            children[Rnd(rnd, geneNum)].Mutation();
            newPop = new Population(children);
            return newPop;
        }
        public Population SortPop()
        {
            Population newPop;
            List<double> fitness = new List<double>();

```

```

        for (int i = 0; i < popSize; i++)
        {
            fitness.Add(population[i].GetFitnessValue());
            //Console.WriteLine(fitness[i].ToString());
        }
        avgFit = fitness.Sum() /Convert.ToDouble( fitness.Count);
        Console.WriteLine(avgFit);
        Chromosome[] tempArr1 = population.ToArray();
        Array.Sort(fitness.ToArray(), tempArr1);
        population = tempArr1.ToList();
        newPop = new Population(population);
        return newPop;
    }
    public bool CheckPop()
    {
        bool flag=false;
        for (int i = 0; i < population.Count; i++)
        {
            if (population[i].GetFitnessValue() > 0.8) flag = true;
        }
        return flag;
    }

    public void ShowPopulation()
    {
        for (int i = 0; i < population.Count; i++)
        {
            System.Console.WriteLine(population[i].GetSpecie());
        }
    }
    public int Rnd(Random r, int p)
    {
        return r.Next(p);
    }
}
}

```

Currently my programm doesn't properly and only gives result after 40000 – 90000 generations so I wont put graph in my report.